FUJII et al.
U. S. Application No. 10/694,822
Our Ref. 8017-1105

Reference Citation List

1. Japanese Unexamined Patent Application 2001-312481
2. Japanese Unexamined Patent Application 2518902

Claims 1-13
Citations 1 and 2

Remarks:

In Citation 2, reference is made to an array processor composed with various connections and in which multiple processor element command codes are chronologically switched by multiple scheduling circuits, with the multiple processor elements being divided into individual element groups corresponding to the multiple scheduling circuits.

The multiple scheduling circuits of the invention recorded in Citation 2 are connected by special communication lines located between the scheduling circuits, with control being accomplished upon receipt of an event notification from a proximate scheduling circuit (Fig. 1 and Fig. 7).

Hence, the invention relating to Claims 1-13 of the present application could be easily conceived by one skilled in the art by applying the technical concept recorded in Citation 2 to the invention recorded in Citation 1.

Record of the Examination Results relating to Documents of the Prior Art

Examined Technical Field: G06F15/80
H03K

Documents of the Prior Art

The record of the examination results relating to documents of the prior art does not constitute the grounds for rejection.

(19)日本国特許庁（ＪＰ）　(12)特　許　公　報（Ｂ２）　(11)特許番号

第2518902号

(45)発行日　平成８年(1996)７月31日　(24)登録日　平成８年(1996)５月17日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ 15/80			Ｇ０６Ｆ 15/80		
15/16	３９０		15/16	３９０Ｔ	

請求項の数２(全 22 頁)

(21)出願番号　特願昭63－234546

(22)出願日　昭和63年(1988)９月19日

(65)公開番号　特開平2－82378

(43)公開日　平成２年(1990)３月22日

(73)特許権者　999999999
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　進藤　達也
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　河村　薫
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　梅田　政信
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　小笠原　吉義　(外２名)

審査官　酒井　恭信

最終頁に続く

(54)【発明の名称】　並列計算機におけるイベントスケジューリング処理方式

(57)【特許請求の範囲】

【請求項１】処理対象となるデータを記憶する回路と演算回路とをそれぞれ有する複数のプロセッサ（14）と、これらのプロセッサを制御するコントローラ（10）とを備え，コントローラから送出する一連の命令によって，各プロセッサに与えられたデータを処理する並列計算機において，

上記プロセッサを，複数個ずつ，アドレスに関する制御単位となる複数の制御グループ（G1,G2,…）にグループ化し，

各制御グループごとに，

隣接する制御グループからの処理対象データ群に対するアドレスを指定するイベントを受信し管理するスケジューリング回路（11）と，

このスケジューリング回路によって管理されるイベントに基づいて決定されるベースアドレスと，上記コントローラから供給されるアドレスとにより，その制御グループに属するプロセッサが処理すべきデータのアドレスを生成する実アドレス生成回路（12）とを備えたことを特徴とする並列計算機におけるイベントスケジューリング処理方式。

【請求項２】上記各制御グループ間の境界部周辺に位置する各プロセッサに対応して，自己が属する制御グループに隣接する制御グループのプロセッサが扱うデータのアドレスに対応するデータを，その隣接する制御グループにおける隣接プロセッサへ送出する擬似隣接プロセッサ（15）を当該自己が属する制御グループ内に備えたことを特徴とする請求項１記載の並列計算機におけるイベントスケジューリング処理方式。

【発明の詳細な説明】

〔概要〕

SIMD型計算機等において，多数のプロセッサを効率よく動作させることができるようにした並列計算機におけるイベントスケジューリング処理方式に関し，

高い負荷分散を得ることができる制御を可能とした並列計算機を提供することを目的とし，

並列計算機によるプロセッサを，複数個ずつ，アドレスに関する制御単位となる複数の制御グループにグループ化し，各制御グループごとに，隣接する制御グループからの処理対象データ群に対するアドレスを指定するイベントを受信し管理するスケジューリング回路と，このスケジューリング回路によって管理されるイベントに基づいて決定されるベースアドレスと，上記コントローラから供給されるアドレスとにより，その制御グループに属するプロセッサが処理すべきデータのアドレスを生成する実アドレス生成回路とを備えるように構成する。

〔産業上の利用分野〕

本発明は，SIMD型計算機等において，多数のプロセッサを効率よく動作させることができるようにした並列計算機におけるイベントスケジューリング処理方式に関する。

例えば、配線処理等のLSI－CADを含む大規模組み合わせ問題は，非常に高速な演算能力を必要とする。これを解決するためには，多数のプロセッサを用いて構成する超並列計算機によって，高速演算処理を行うことが有望であり，多数のプロセッサを効率よく動作させる技術が必要となる。

〔従来の技術〕

並列計算機の構成として，複数のプロセッサが，個別の命令によってそれぞれ与えられたデータを処理するMIMD（Multiple Instruction stream Multiple Data stream）型のものや，すべてのプロセッサが，同一の命令の流れによってそれぞれ与えられたデータを処理するSIMD（Single Instruction stream Multiple Data stream）型のものなどがある。

プロセッサ数が数十～数百のシステムでは，MIDM型の構成をとることによって，それぞれプロセッサが処理すべきデータを独立に扱い，負荷分散を上げることができる。しかしながら，MIMD型の並列計算機では，プロセッサごとに制御回路を用意しなければならないので，物量が多くなり，例えば数万規模の構成をとることは，現実的に不可能である。

一方，SIMD型の並列計算機では，数万規模の構成のものが実現されている（例：Thinking Machines社のコネクション・マシン）。

SIMD型の並列計算機では，同一の命令の流れで，各プロセッサを制御するので，非常に多数のプロセッサを持つ構成を実現することができる。

〔発明が解決しようとする課題〕

SIMD型の場合，多数のプロセッサを持つ並列計算機を実現できる反面，すべてのプロセッサに対し、同じ動作をさせることになるため，負荷分散が難しいという問題がある。例えば，SIMD型の並列計算機を，LSI設計における配線処理に用いた場合，迷路法におけるウェーブフロント等を担当するプロセッサは，ほんの一部であり，大部分のプロセッサは，実質的に休止の状態となる。このような応用分野に限らず，SIMD型の並列計算機では，多くのプロセッサを無駄なく動作させるということが困難である。

本発明は上記問題点の解決を図り，高い負荷分散を得ることができる制御を可能とした並列計算機を提供することを目的としている。また，各プロセッサが処理するデータの連携を，効率的に行う手段を提供することを目的としている。

〔課題を解決するための手段〕

第１図は本発明の構成例を示す。

第１図において，10はコントローラ，11はイベントのスケジューリングを行うスケジューリング回路，12は実アドレスを生成する実アドレス生成回路，14はプロセッサ（PE：Processing Element），15は擬似隣接プロセッサ，G1～G4はアドレスに関する制御単位となる制御グループを表す。

プロセッサ14は，各々，処理対象となるデータを記憶する回路と演算回路とを持ち，コントローラ10から送られてくる制御信号による同一の命令の流れによって，各プロセッサ14に与えられたデータを処理する。処理対象となるデータは，各プロセッサ14に個別に用意されたメモリに格納され，コントローラ10からのアドレス信号によって，そのデータが読み・書きされる。

本発明では，プロセッサ14は，複数個ずつ，アドレスに関する制御単位となる複数の制御グループG1，G2，…にグループ化される。そして，各制御グループG1，G2，…ごとに，スケジューリング回路11と実アドレス生成回路12とが設けられる。

スケジューリング回路11は，隣接する制御グループからの処理対象データ群に対するアドレスを指定するイベントを受信し，そのイベントによって指定されたアドレスをキュー等によって管理する回路である。

実アドレス生成回路12は，スケジューリング回路11によって管理されるイベントに基づいて決定されるベースアドレスと，コントローラ10から供給されるアドレスとにより，その制御グループに属するプロセッサ14が処理すべきデータのメモリ上における実際のアドレスを生成する回路である。

また，各制御グループG1，G2，…間の境界部周辺に，必要に応じて，擬似隣接プロセッサ15が設けられる。擬似隣接プロセッサ15は，各制御グループ間の境界部周辺に位置するプロセッサ14が，隣接する制御グループにおける隣接プロセッサ14とデータの授受を行うときに，あたかも処理対象領域が連続しているかのように，それぞれ

のプロセッサが扱うデータのアドレスに対応するデータを送出する機構を持つ。

〔作用〕

従来のSIMD型計算機等では、各プロセッサは同一の命令の流れで制御され，その各プロセッサが持つメモリのアドレスも同一のものが供給されるため，処理対象領域は単一的である。

本発明では，プロセッサ14が，いくつかの制御グループに分割され，各制御グループごとに，異なる領域を，処理対象として扱うことができる構成になっている。すなわち，各制御グループごとに，スケジューリング回路11が指定するベースアドレスと，コントローラ10からの相対アドレスとによって，実際にデータが存在するアドレスが生成されるようになっている。従って，各プロセッサ14における処理対象領域を，各制御グループにおいて実際に処理が必要なデータが存在する個所に持っていくことができる。処理が必要であるかどうかは，隣接する制御グループ間で通知されるイベントを管理することによって判別する。

実装されているプロセッサ14と，それが扱う処理対象データのアドレスとが，各制御グループごとに可変化されて制御されるので，処理対象データに着目して見た場合，1つのプロセッサ14上で，複数の仮想的なプロセッサが動作しているかのように見ることができる。このように，仮想プロセッサとイベントの概念を導入することにより，プロセッサ14の負荷分散が可能となる。

処理対象領域の連続性を保証する必要がある場合には，各制御グループ間の境界部周辺に，擬似隣接プロセッサ15を設けることにより，アドレスの調整が可能である。擬似隣接プロセッサ15がない場合，制御グループ間でアドレスを調整し，相手が必要とするデータを送るための制御フェーズを設ければよいが，その分だけ効率が悪くなる。

〔実施例〕

第２図は本発明の実施例で用いられるプロセッサの構成例，第３図は本発明の実施例に係る仮想プロセッサの概念を説明する図，第４図は本発明の実施例に係るウィンドウ分割説明図，第５図は本発明の実施例に係るアドレス割り付けの例，第６図は本発明の実施例に係る制御グループの例，第７図は本発明の実施例に係るスケジューリング回路接続例，第８図は本発明の実施例に係る擬似隣接プロセッサ説明図，第９図は本発明の実施例に係るスケジューリング回路ブロック図，第10図は第９図に示すウィンドウ番号入力部の例，第11図は第９図に示す連続性検出部の例，第12図は第９図に示すイベント入力部の例，第13図は第９図に示すイベント解釈回路論理説明図，第14図は第９図に示すFIFO回路の例，第15図は第９図に示す登録フラグの列，第16図は第９図に示すアドレス計算回路説明図，第17図は本発明の実施例に係る実アドレス生成回路の例を示す。

本実施例の並列計算機は，第２図に示すようなプロセッサ14を格子状に結合した構成になっている。以下，格子結合されたマルチプロセッサを例に説明するが，本発明は，格子結合に限らず，超立方体結合やその他の結合による並列計算機にも，同様に適用することができる。

各プロセッサ14は，内部に，演算対象となるデータを保持するデータレジスタ21と，入出力データおよびデータレジスタ21に記憶されているデータについての演算を行う演算回路（ALU）を持つ。また，データレジスタ21へのロード対象となるデータを記憶する外部メモリ20を持つ。外部メモリ20のアドレスは，各制御グループごとに設けられた実アドレス生成回路12から供給される。本発明は，主として，この外部メモリ20に対するアドレス制御に関連している。

各プロセッサ14は，第１図に示すコントローラ10から送られる同一の制御信号で制御される。この制御信号には，データレジスタ21のアドレス，演算回路22に対するオペレーション・コード等が含まれる。

プロセッサ14は，東（E），西（W），南（S），北（N）の４方向の隣接プロセッサとの通信を行うためのEWNSポートを持つ。また，全プロセッサ14に対して，コントローラ10から同じ値を与えるためのグローバルデータ入力の端子と，コントローラ10へのデータ出力の端子を持つ。

第２図に示すプロセッサ14は，1ビット・プロセッサであり，データレジスタ21への入出力や外部とのデータ入出力は，すべて１ビットを基本としている。１ビットよりも大きいデータは，その大きさに応じて１ビットずつ，上位ビットまたは下位ビット側から連続して処理する。これにより，何ビットのデータでも処理することができるようになっている。もちろん，本発明は，この例のような１ビット・プロセッサに限らず，複数ビットの入出力を行うプロセッサにも同様に適用することが可能である。

本発明では，並列計算機上に実装されているプロセッサ14の数より，多数のプロセッサがあるかのように取り扱い可能とするため，仮想プロセッサの概念を導入する。例えば，LSI設計における配線処理を並列計算機によって行う場合，プロセッサ数よりも大きな配線領域を扱うことができることが必要となり，そのような配線処理のプログラムを，実際のプロセッサ数を気にしないで記述できるようにすることが望まれる。

そのため，第３図に示すように，実装されているプロセッサが一時点で直接処理対象とする領域よりも大きな二次元のメモリ空間30内を，実プロセッサ（PE）群が移動して処理していくことで，メモリ空間30の大きさを持つ仮想PE群があるかのように見せる。

具体的には，第４図に示すように，メモリ空間30の仮想領域について，例えば128×128個の仮想PEの大きさを単位とするウィンドウ31を，最大256個用いて管理す

る。すなわち，仮想領域を，n×mのウィンドウ31による任意な短形で組み合わせて管理する。各ウィンドウ31は，第４図に示すように，0からnm－１までのウィンドウ番号によって識別する。

１個の実PEに着目してみると，第５図（イ）に示すように，1PEの外部メモリ20を分割し，その１つ１つを仮想PEのメモリ空間として利用することになる。この例では，実PEのメモリ空間が，0000番地からFFFF番地までの16ビットのアドレス空間を持ち，仮想PEのメモリ空間を，それぞれ12ビットのアドレス空間としている。１個の実PEが16個の仮想PEを担当することになる。

外部メモリ20に対するアドレスは，第５図（ロ）に示すように生成される。ウィンドウ番号は，個々の仮想PEのメモリ空間の先頭を示すベースアドレスとなる。ウィンドウは，最大256個設けることができるため，ウィンドウ番号として８ビット用意されるが，この例のように，16分割した場合には，ウィンドウ番号の下位４ビットを“0”とする。

仮想PEアドレスは，各仮想PEのメモリ空間における相対アドレスである。この仮想PEアドレスは，処理対象データへのアクセスが必要な場合に，第１図に示すコントローラ10から，各プロセッサ14に対して共通に送られるアドレスである。仮想PEアドレスは，ウィンドウの個数に応じて，その上位ビットに“0”が詰められる。16個のウィンドウに分割されている場合，実質的な仮想PEアドレスは12ビットである。

第５図（ロ）に示すように，ウィンドウ番号と仮想PEアドレスとを加算する，または論理和をとることにより，外部メモリ20に対する実アドレスが生成されることになる。

仮想PEにおける処理の実行は，担当する実PEが，分割されたメモリ空間内のデータを順に処理していくことで行う。この最も単純な実現法として，実PEが，常に，担当するすべての仮想PEを順に処理していくことが考えられる。しかし，仮想PEの中で，本当に処理が必要なものだけを抽出して実行させることができるならば、その部分だけを選択的に実行制御することにより，処理の効率化を図ることが可能である。そのため，イベントの概念を導入し，無駄な仮想PEの処理を省く制御を行う。

イベントは，仮想PEの処理すべき条件が成立したときに起動されるものである。イベントが伝播された仮想PEは，実PEにおける処理の対象として扱われる。イベントの例として，配線処理においては，ラベリング時にラベルの値が書き換わること，バックトレース時にトレース信号が伝播されることなどが挙げられる。どちらも配線領域内において，それらが発生した部分についての処理（ラベル値の評価，トレース処理）を行うべき事象である。何をイベントとして扱うかは，各プロセッサ14を制御するコントローラ10が，プログラムに応じて決定する。

次に，以上のような仮想PEの具体的な実現のし方について説明する。

まず，第６図に示すように，プロセッサ14群を，制御グループと呼ぶ短形の集合に分ける。第６図に示す例では，128×128個のプロセッサ14を，32×32個のプロセッサ14を持つ16個の制御グループG1, G2, …に分割した構成になっている。

この各制御グループG1, G2, …ごとに，仮想PE上の担当する領域の中から，イベントの起こったところを次々に処理していく。従って，制御グループ内では，連続した領域を扱うことになるが，制御グループ間の境界では，必ずしも連続した領域とはならない。このようにPE全体ではなく，それを分割した単位で，任意の領域を処理できるようにすることで，各プロセッサ14の稼動率を上げる。稼動率の点だけを考えると，制御グループを構成するPE数を小さくしたほうが好ましいが，その実現に必要な物量が増加することになる。

上述したイベントの管理のために，第７図に示すように，各制御グループG1, G2, …ごとに，スケジューリング回路11を設け，また実アドレス生成回路12を設ける。

スケジューリング回路11は，各制御グループのPE群からイベントを受信し，処理すべき仮想PEを管理する。イベントにより処理対象となった仮想PEの番号，すなわちウィンドウ番号は，スケジューリング回路11においてキューイングされ，キューの先頭から順に処理される。

処理実行の順がまわってきた仮想PEの処理は，スケジューリング回路11が，その仮想PEに対応するベースアドレスを，実アドレス生成回路12に送ることにより行われる。このように，スケジューリング回路11が行うべき処理は，イベントの伝播した仮想PEのキューイングと，それらに対する実PEに割り付けである。

実アドレス生成回路12は，コントローラから全PEに対し共通に送られてくる制御信号の１つである仮想PEの相対アドレスと，スケジューリング回路11が決定した仮想PEのベースアドレスとから，実アドレスを生成し，各制御グループG1, G2, …に存在する実PEへ供給する。

スケジューリング回路11は，それぞれ隣接する４方向の制御グループに属するPE群およびその隣接制御グループのスケジューリング回路11と，自分の担当する制御グループ内のPE群とに接続される。主な入出力信号は，以下のとおりである。

（ａ）イベント信号〔入力〕

隣接する制御グループ境界上のPEから伝えられるイベント信号（全部で32PE分）のオア（OR）論理をとったもので，E・W・N・Sの各方向に対して，各１ビットの入力信号である。

（ｂ）ウィンドウ番号〔入力〕

隣接するスケジューリング回路11のウィンドウ番号出力が入力される。各方向に対して８ビットの入力信号である。イベント信号がアクティブになったときに，その

方向のウィンドウ番号を入力しキューイングする。

（ｃ）自己（self）イベント信号〔入力〕

自分が担当する制御グループ内PEのイベント信号（全部で32×32PE分）のOR論理をとった１ビットの入力信号である。

（ｄ）ウィンドウ番号〔出力〕

隣接するスケジューリング回路11へのウィンドウ番号出力で，各方向ごとに８ビットである。

（ｅ）ベースアドレス〔出力〕

キューの先端から取り出された仮想PEのウィンドウ番号に対応するアドレスを示す実アドレス生成回路12への出力信号である。

（ｆ）各種制御信号〔入力，出力〕

コントローラからの入力またはコントローラへの出力信号であり，次の仮想PEをキューから取り出すことを指示する制御信号（next）入力，データの流れる東西南北の方向を示す制御信号（dir）入力，クロック信号入力，キューが空になったことを示す制御信号（empty）出力等がある。

次に，制御グループ間の領域の連続性を効率よく保証するために用いる擬似隣接プロセッサについて，第８図に従って説明する。

第８図において，14A，14Bは制御グループの境界を越えて互いに隣接するプロセッサである。このプロセッサ14A，14Bに対応して，擬似隣接プロセッサ1A，15Bが設けられる。

隣接する制御グループが，異なるウィンドウを処理対象としているとき，制御グループの境界を越えて，隣接するPEの値をそのまま受け取ると，必要とするウィンドウ内の値を受け取ることができなくなる。

そのため，第８図に示すように，制御グループの境界部における１接続ごとに，相手側のウィンドウに相当するデータを送り出すためのPEを余分に置き，擬似隣接プロセッサ15A，15Bとする。この擬似隣接プロセッサ15A，15Bを設けることにより，扱う領域が不連続となる制御グループ間の境界部においても，隣接するPEの値を用いた計算の実行を保証することができる。扱うウィンドウが，隣接する制御グループ間で連続している場合には，セレクタS1，S2を切り換えることにより，擬似隣接プロセッサ15A，15Bを用いないで，直接，隣接するプロセッサ14A，14B間でアクセスする。

すなわち，プロセッサ14Aは，自分の外部メモリ20Aに対して，リード／ライト（R/W）するとき，ライト・データについては，擬似隣接プロセッサ15Aの外部メモリ20aにも同時に書き込む。プロセッサ14Bへ，プロセッサ14A側からデータを送り出す場合，擬似隣接プロセッサ15Aが，外部メモリ20aの値を読み出して，プロセッサ14Aの代わりに，その値を送信する。その外部メモリ20aのアドレスは，プロセッサ14B側が処理しているウィンドウのアドレスとする。プロセッサ14B側からプロセッサ14Aへ値を送る場合も同様である。

第８図では，1次元方向の接続だけを示しているが，実際には，2次元格子における制御グループの境界部において，2次元方向に同様の接続を行う。

本実施例に係るスケジューリング回路は，第９図に示すような構成になっている。

第９図において，50はウィンドウ番号入力部，51は登録テーブル，52は連続性検出部，53はイベント入力部，54はイベント解釈回路，55はFIFO（First In First Out）回路，56は登録フラグ，57はアドレス保持フラグ，58はアドレス計算回路，R1～R4はパイプライン制御のためのレジスタである。

第９図に示すウィンドウ番号入力部50は，隣接する４方向の制御グループから通知されるウィンドウ番号を入力する回路であり，第10図に示すような構成になっている。第10図に示す各レジスタR10は，東西南北（EWNS）の各方向からのウィンドウ番号を保持するレジスタである。セレクタS10は，プロセッサ間通信の方向を示すコントローラからの制御信号dirに応じて，出力を選択する。

第９図に示す登録テーブル51は，ウィンドウ番号が登録されているかどうかを示すフラグ群からなるテーブルであり，最大256個の各ウィンドウに対して，それぞれ１ビットが割り当てられている。ウィンドウ番号入力部50からのウィンドウ番号が，登録テーブル51のアドレスとなる。これによって，ウィンドウ番号の二重登録が抑止されるようになっている。

第９図に示す連続性検出部52は，現在扱っている領域が，隣接する制御グループの領域と連続しているかどうかを判定するための回路である。内部構成は，第11図に示すようになっている。

第11図において，COMPは比較回路，60はエンコーダ，ORはオア回路，CE，CW，CN，CSはリセット信号resetがくるまで，連続性の検出結果を記憶するレジスタである。

自制御グループで現在扱っているアドレスの上位ビット（bits）と，ウィンドウ番号入力部50からの入力ウィンドウ番号とを，比較回路COMPで比較し，それが一致するときに，EWNの方向に応じて，レジスタCE，CW，CN，CSに連続性を記憶する。その結果は，セレクタS20を介して出力される。

第９図に示すイベント入力部53は，各方向からのイベント信号を入力する回路であり，第12図に示すような構成になっている。各レジスタＲは，イベント・クリア信号evclrによってクリアされる。その後，イベント信号がロードされると，アンド回路ANDおよびセレクタS30を介して，制御信号dirに応じたイベント信号が出力される。第９図に示すイベント解釈回路54は，ウィンドウ番号をキューイングするかどうか，また現在のアドレスを保持し続けるかどうかを判断する回路である。第13図に示すような論理で出力を決定する。ウィンドウ番号のキ

ューイングを指示する登録信号は，イベント入力部53の出力だけがアクティブ（T）になったときに出力される。連続性出力およびイベント入力がアクティブになったとき，または自己イベントがアクティブになったとき，アドレス保持信号が出力される。

第９図に示すFIFO回路55は，伝播されたイベントにより，処理しなければならないウィンドウの番号を記憶しておく回路である。通常用いられているFIFOメモリでよく，例えば第14図に示すような構成になっている。

第14図において，MEMは８ビット×256の容量を持つメモリ，R40～R43はレジスタ，S40はセレクタ，WCNTは書き込みアドレスを出力するライトカウンタ，RCNTは読み出しアドレスを出力するリードカウンタ，COMPは比較回路，ORはオア回路，ANDはアンド回路，NOTはノット回路である。

レジスタR41に登録信号がセットされると，所定のタイミングで，レジスタR40にセットされたウィンドウ番号が，ライトカウンタWCNTに示されるメモリMEMのアドレスに書き込まれる。また，読み出しを指示する制御信号nextにより，リードカウンタRCNTの示すアドレスのメモリMEMの内容が読み出されて，レジスタR4を介して出力される。

ライトカウンタWCNTとリードカウンタRCNTの値が一致したとき，空を示す信号emptyが出力される。

第９図に示す登録フラグ56は，第15図に示すような構成になっており，どの方向からのウィンドウ番号が登録されたかを，レジスタＲに記憶する。

第９図に示すアドレス計算回路58は，FIFO回路55から読み出さたウィンドウ番号に基づいて，隣接する制御グループに通知するウィンドウ番号および実アドレスの生成に使用するアドレス上位ビットを出力する回路である。

隣接制御グループ間インターフェースとして，隣接制御グループにイベントを伝えたときに，伝えられた先が登録すべきウィンドウ番号を出力する。ウィンドウ境界以外では，現在担当しているウィンドウ番号を送る。

ウィンドウ境界では，第16図（イ）に示すように，水平方向には，ウィンドウ番号Ａ±１を送り，垂直方向には，ウィンドウ番号Ａ±Ｂを送る。ここで，Bは仮想領域の大きさをウィンドウを単位として分割した場合の横方向のウィンドウ数である。なお，仮想領域の境界では，それより外にイベントが伝わらないように，イベントを打ち消す。

ウィンドウの境界は，第16図（ロ）に示す各方向別の境界印80によって識別する。境界印80の値は，初期設定時に，コントローラによって設定される。

アドレス計算回路58の概要構成は，第16図（ハ）に示すようになっており，演算回路ALUは，第16図に示す境界印80の値によって，ウィンドウ番号A，A±1，A±Ｂのいずれかを算出する。

なお，コントローラから送られてくるアドレスを，現在のウィンドウ番号に関係なく，絶対アドレスとして使用するモードを持つ。これが，第16図（ハ）の入力の１つであるアドレス指定値であり，そのオペレーションが指示された場合には，そのアドレス指定値が，セレクタS80，S82を介して，実アドレス生成回路へ送られる。これにより，メモリ内に仮想PE間の共通領域を実現することなどが可能になっている。

第１図に示す実アドレス生成回路12は，本実施例では，第17図に示すような構成になっている。第17図において，R100～R105はレジスタ，S100～S103はセレクタ，ORはオア回路である。

実アドレス生成回路の入力は，コントローラから送られてくる仮想PEの相対アドレスと，第９図に示すアドレス計算回路58の出力であるアドレス上位ビットと，ウィンドウ番号入力部50からの隣接グループウィンドウ番号である。

自分の制御グループ内に属するPEに対する実アドレスは，レジスタR100に設定された相対アドレスと，レジスタR101に設定されたアドレス上位ビットとを加算することにより生成する。すなわち，第５図（ロ）に示す演算を行う。ここでは，上位８ビットで重なりあう部分は，一方を“0”とすることとし，オア回路ORによる論理和で加算を実現している。なお，実アドレスの下位８ビットは，コントローラから送られてきたものをそのまま使用する。

また，第８図に示す擬似隣接プロセッサ用の実アドレスを生成するために，レジスタR102～R105に隣接グループのウィンドウ番号を設定し，セレクタS100～S103によって，ロード（Ｌ）時には隣接グループのアドレス，セーブ（Ｓ）時には自分のアドレス（self）と同じになるように制御する。

本実施例で説明した１つの制御グループ，スケジューリング回路11，実アドレス生成回路12を個別に，またはまとめてLSI化することが可能である。スケジューリング回路11および実アドレス生成回路12の詳細な例を示したが，同様な機能を持つものを他の回路構成によっても，実現できることは言うまでもない。また，格子結合のマルチプロセッサを例に説明したが，制御グループによるグループ化は，超立方体結合などの他の結合によるマルチプロセッサでも同様に実現できることは明らかである。

〔発明の効果〕

以上説明したように，本発明によれば，実プロセッサが，多数の仮想プロセッサを担当し，仮想プロセッサの中で真に処理を必要とするものを抽出して，実プロセッサによる処理を遂行することができるので，高い負荷分散が可能になり，処理の効率化が可能になる。

【図面の簡単な説明】

第１図は本発明の構成例，

第２図は本発明の実施例で用いられるプロセッサの構成

13

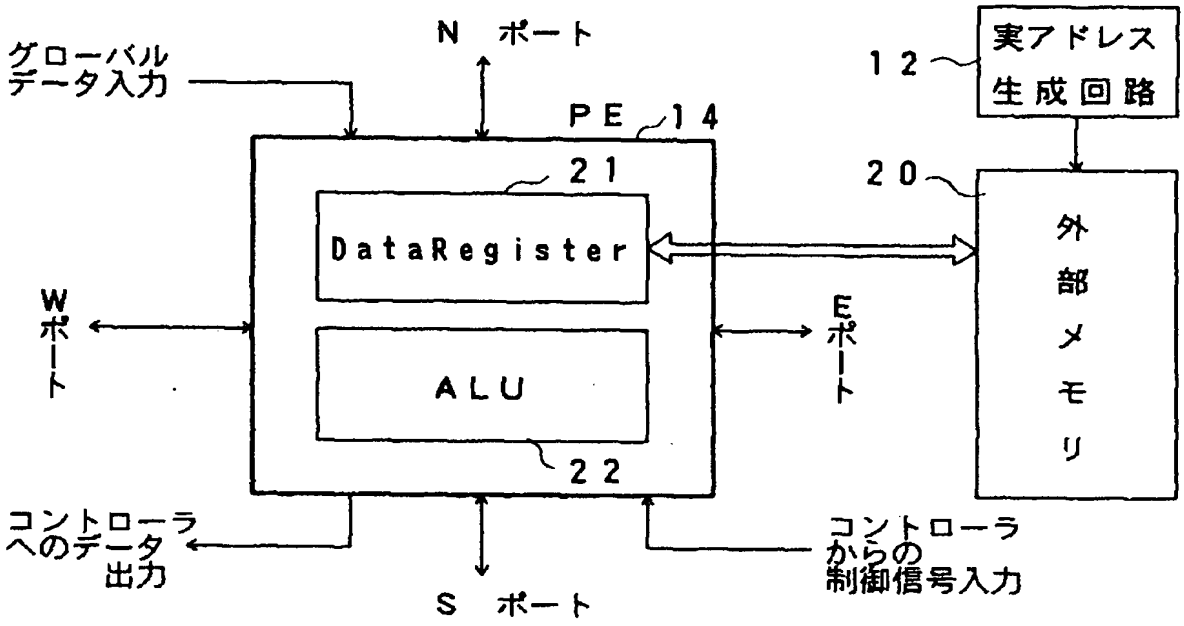
例，
第３図は本発明の実施例に係る仮想プロセッサの概念を説明する図，
第４図は本発明の実施例に係るウィンドウ分割説明図，
第５図は本発明の実施例に係るアドレス割り付けの例，
第６図は本発明の実施例に係る制御グループの例，
第７図は本発明の実施例に係るスケジューリング回路接続例，
第８図は本発明の実施例に係る擬似隣接プロセッサ説明図，
第９図は本発明の実施例に係るスケジューリング回路ブロック図，

14

第10図は第９図に示すウィンドウ番号入力部の例，
第11図は第９図に示す連続性検出部の例，
第12図は第９図に示すイベント入力部の例，
第13図は第９図に示すイベント解釈回路論理説明図，
第14図は第９図に示すFIFO回路の例，
第15図は第９図に示す登録フラグの列，
第16図は第９図に示すアドレス計算回路説明図，
第17図は本発明の実施例に係る実アドレス生成回路の例を示す。

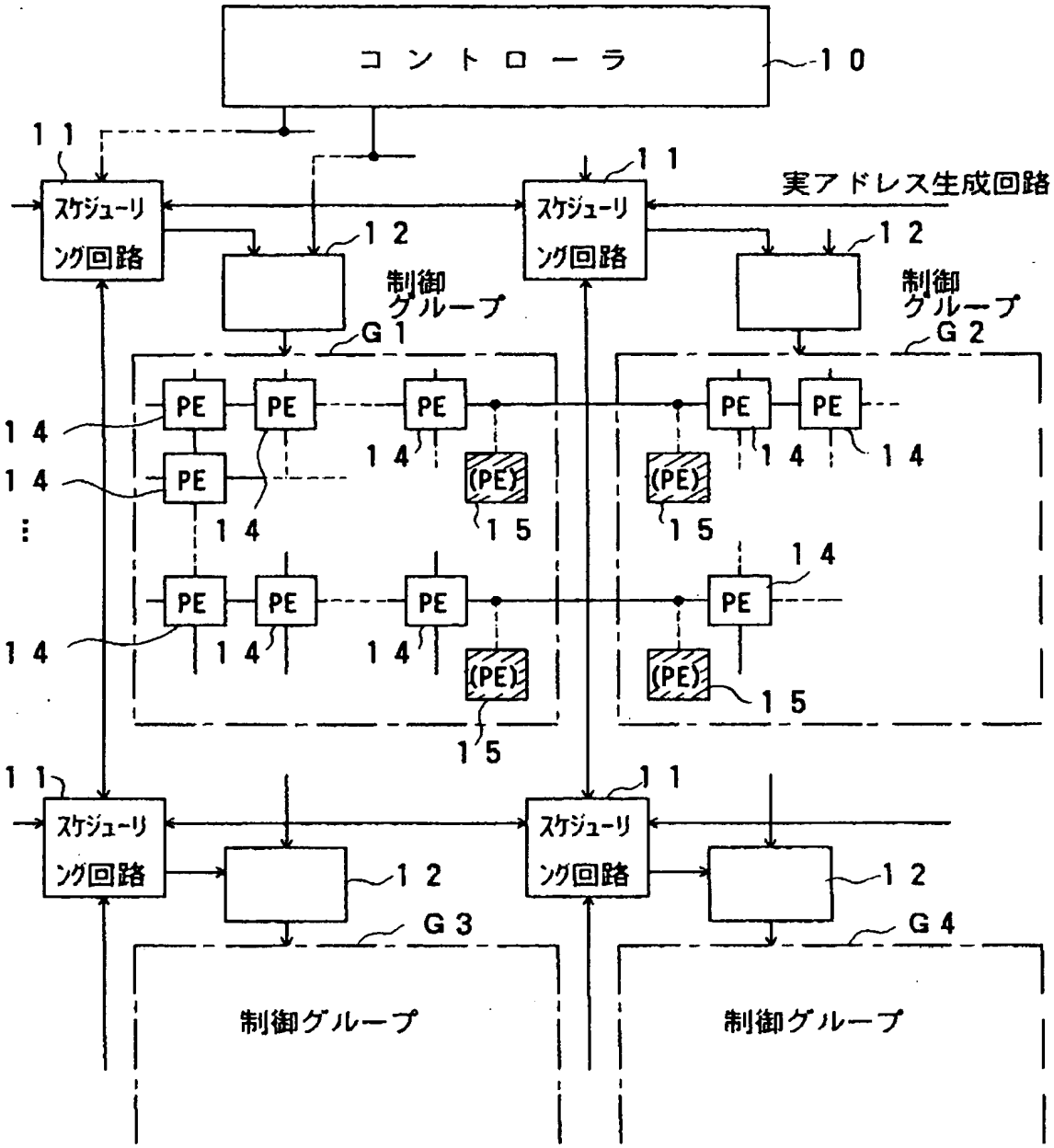
10 図中，10はコントローラ，11はスケジューリング回路，12は実アドレス生成回路，G1，G2，…は制御グループ，14はプロセッサ，15は擬似隣接プロセッサを表す。

【第２図】



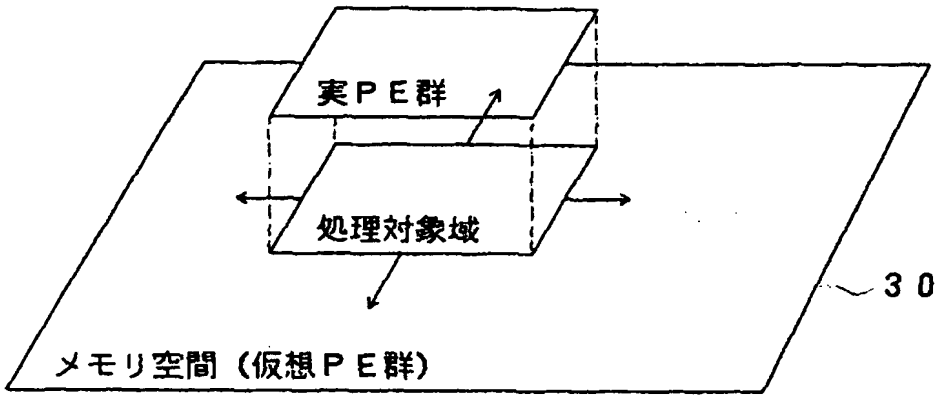
プロセッサの構成例

【第１図】


コントローラ
10
11
スケジューリング回路
12
制御グループ
G1
G2
G3
G4
実アドレス生成回路
PE
(PE)
14
15
制御グループ

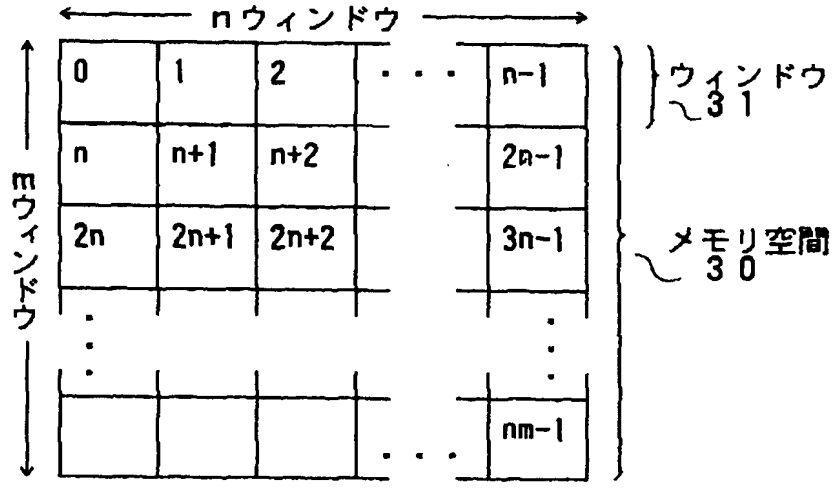
本　発　明　の　構　成　例

【第３図】



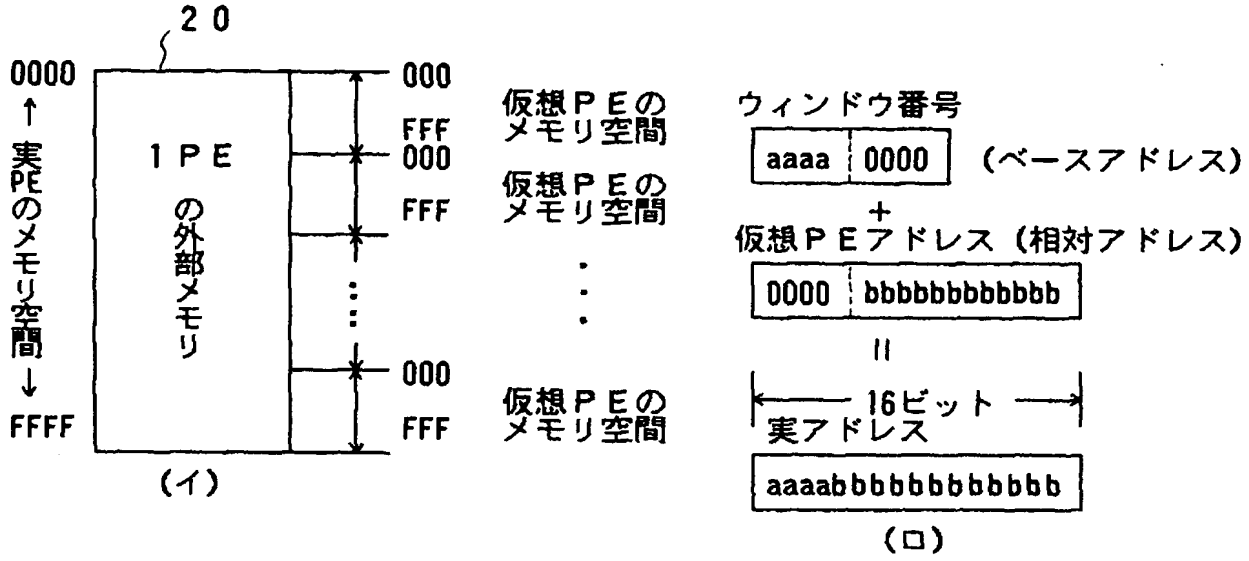
仮想プロセッサの概念説明図

【第４図】



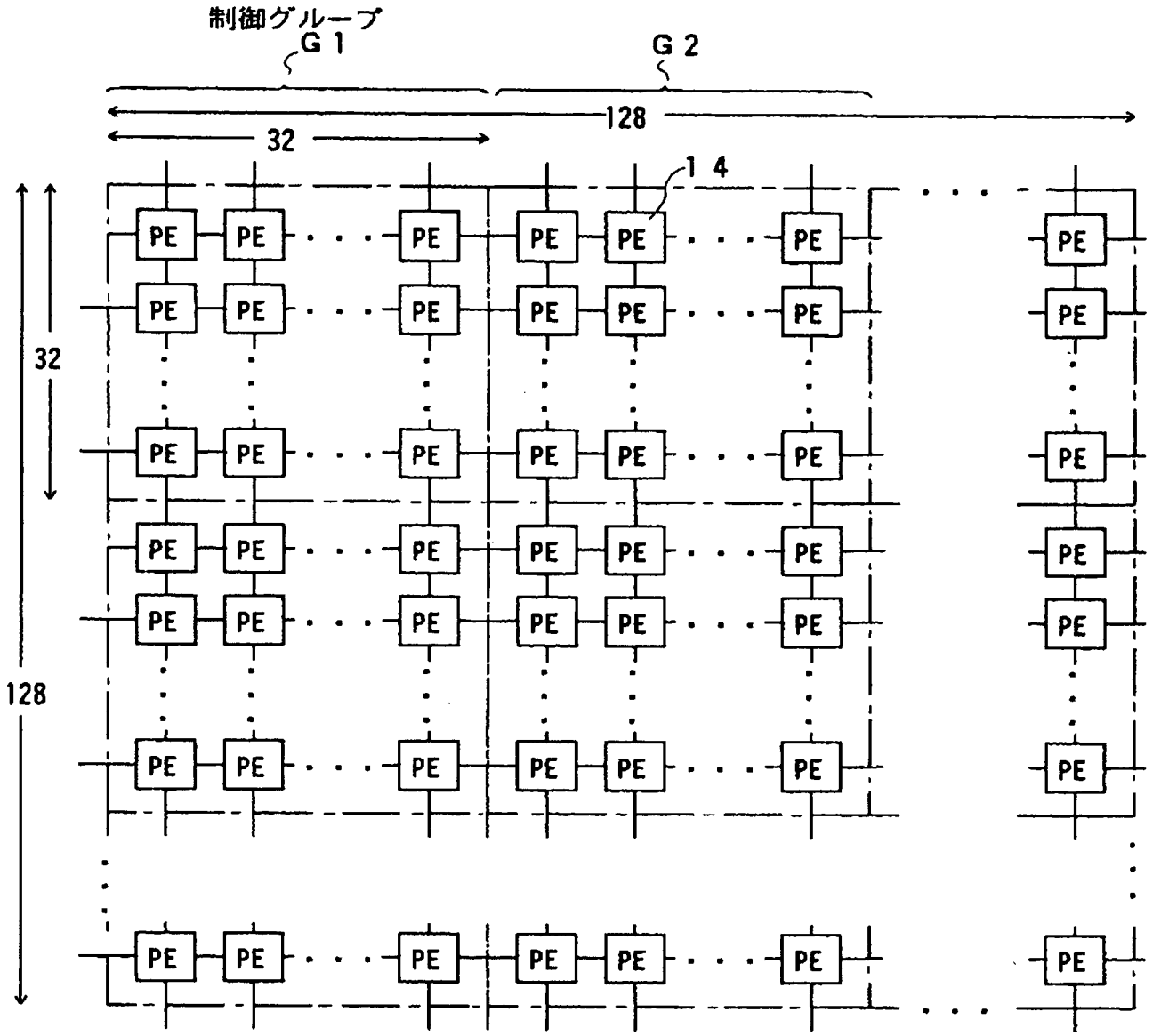
ウィンドウ分割説明図

【第５図】



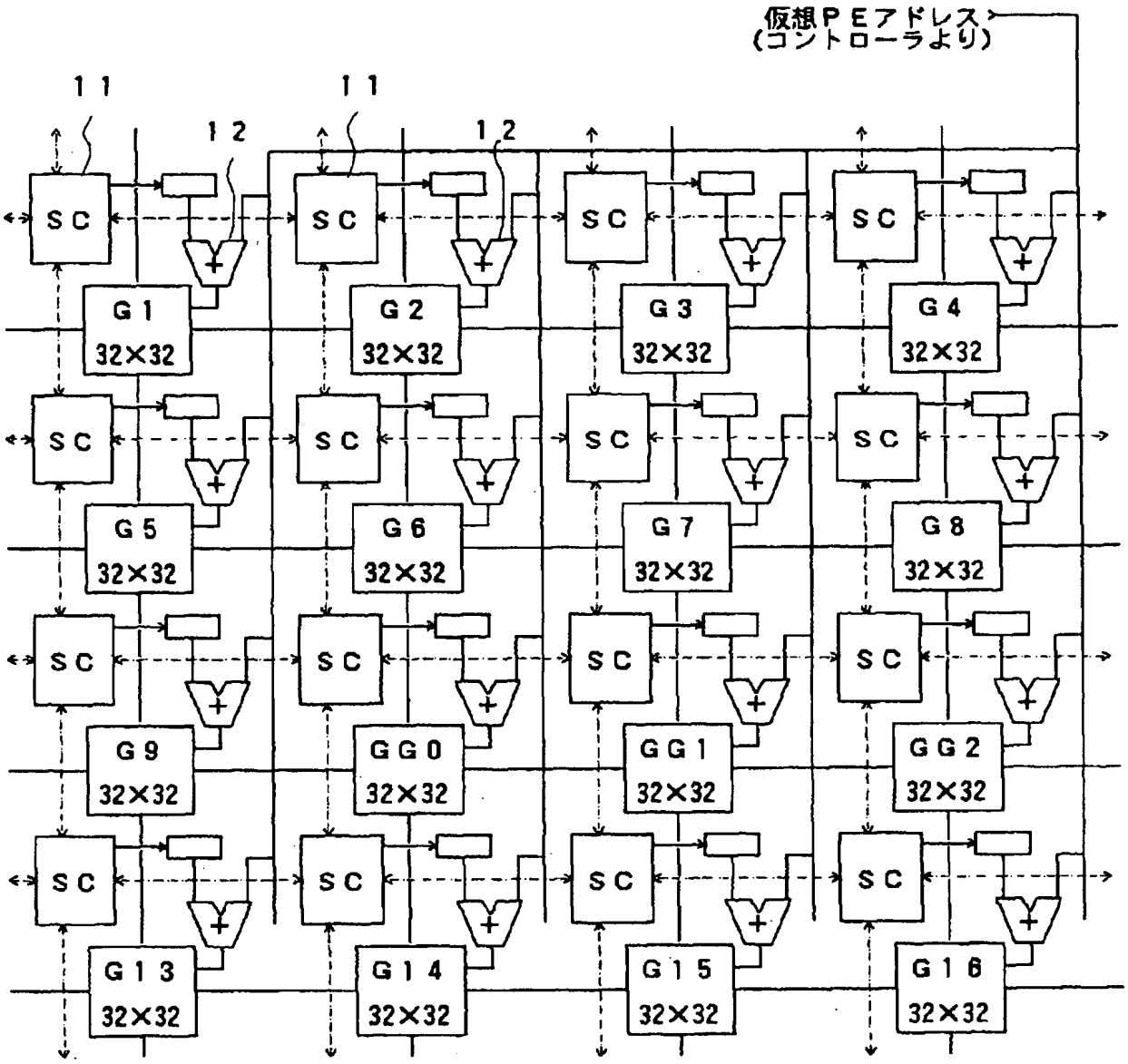
アドレス割り付けの例

【第６図】



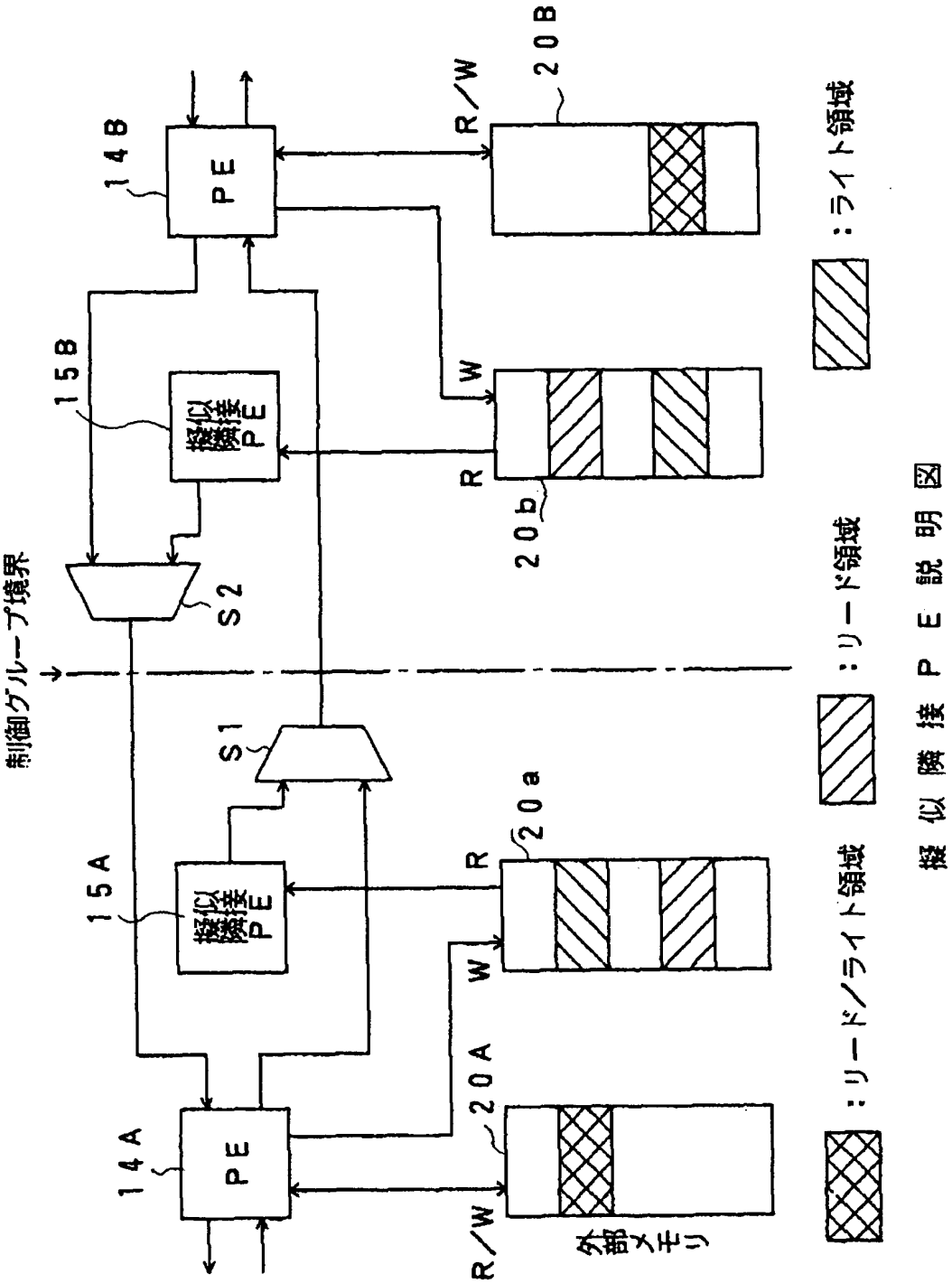
制 御 グ ル ー プ の 例

【第７図】



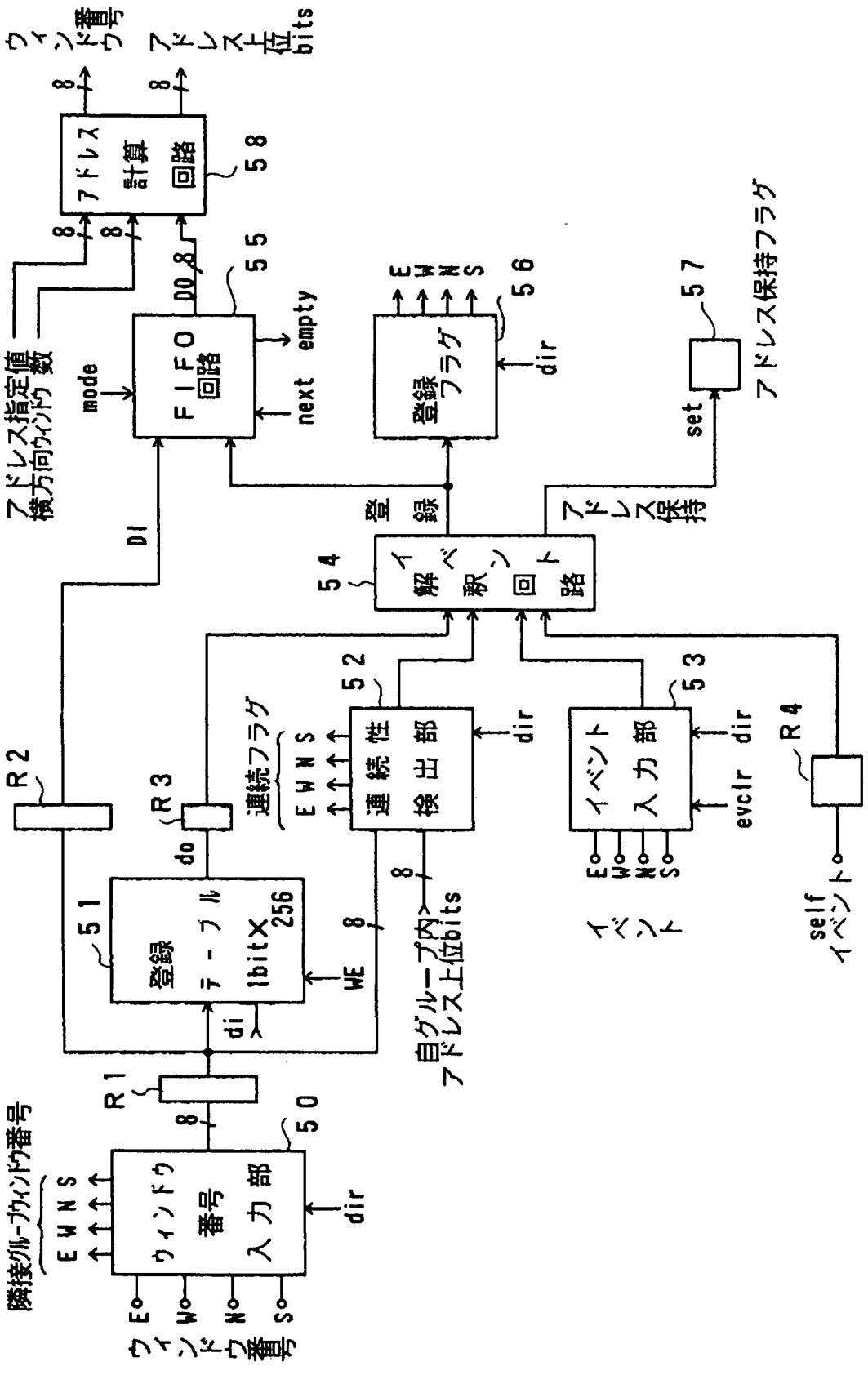
スケジューリング回路接続例

【第８図】



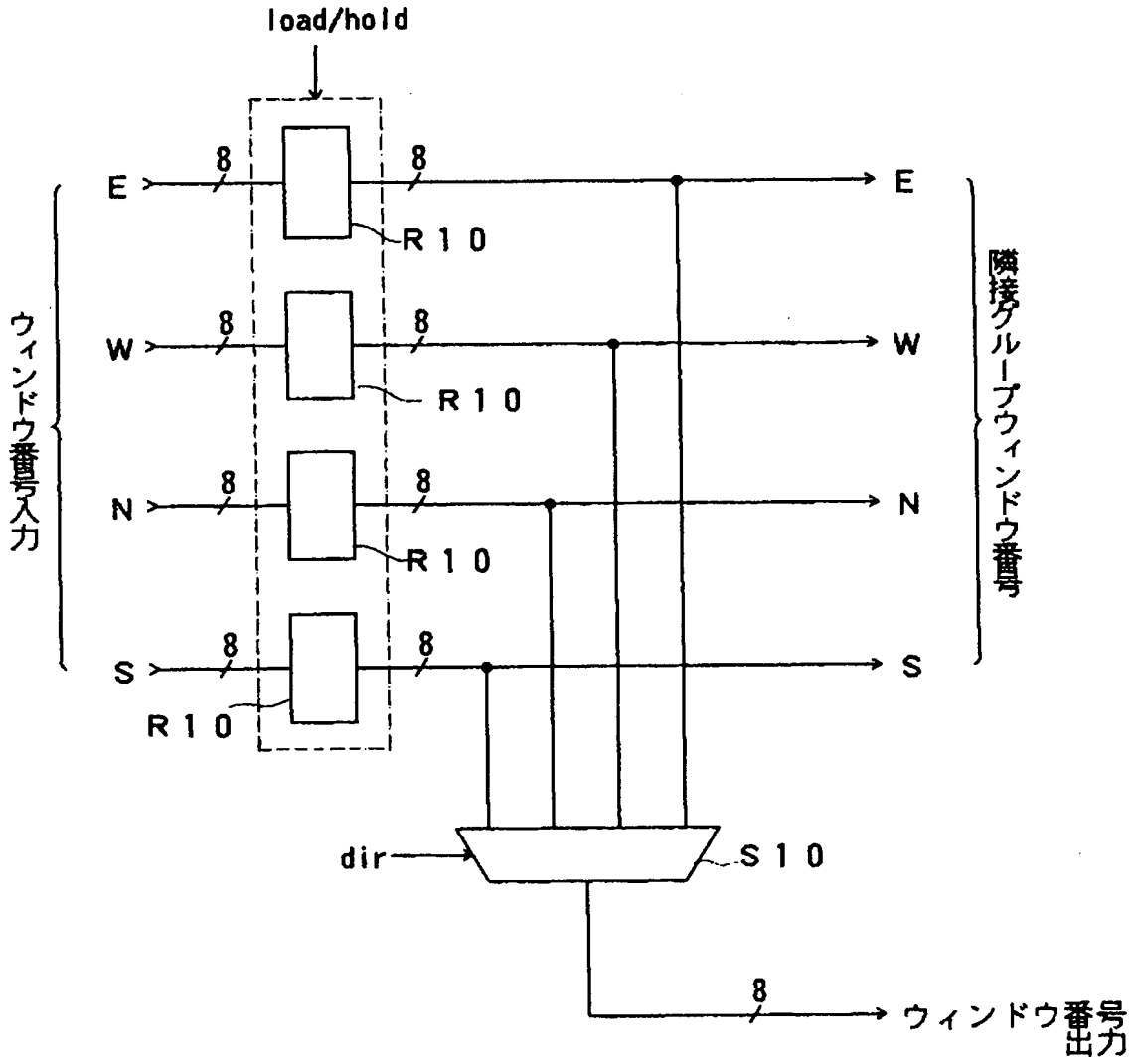
擬似隣接ＰＥ説明図

【第９図】



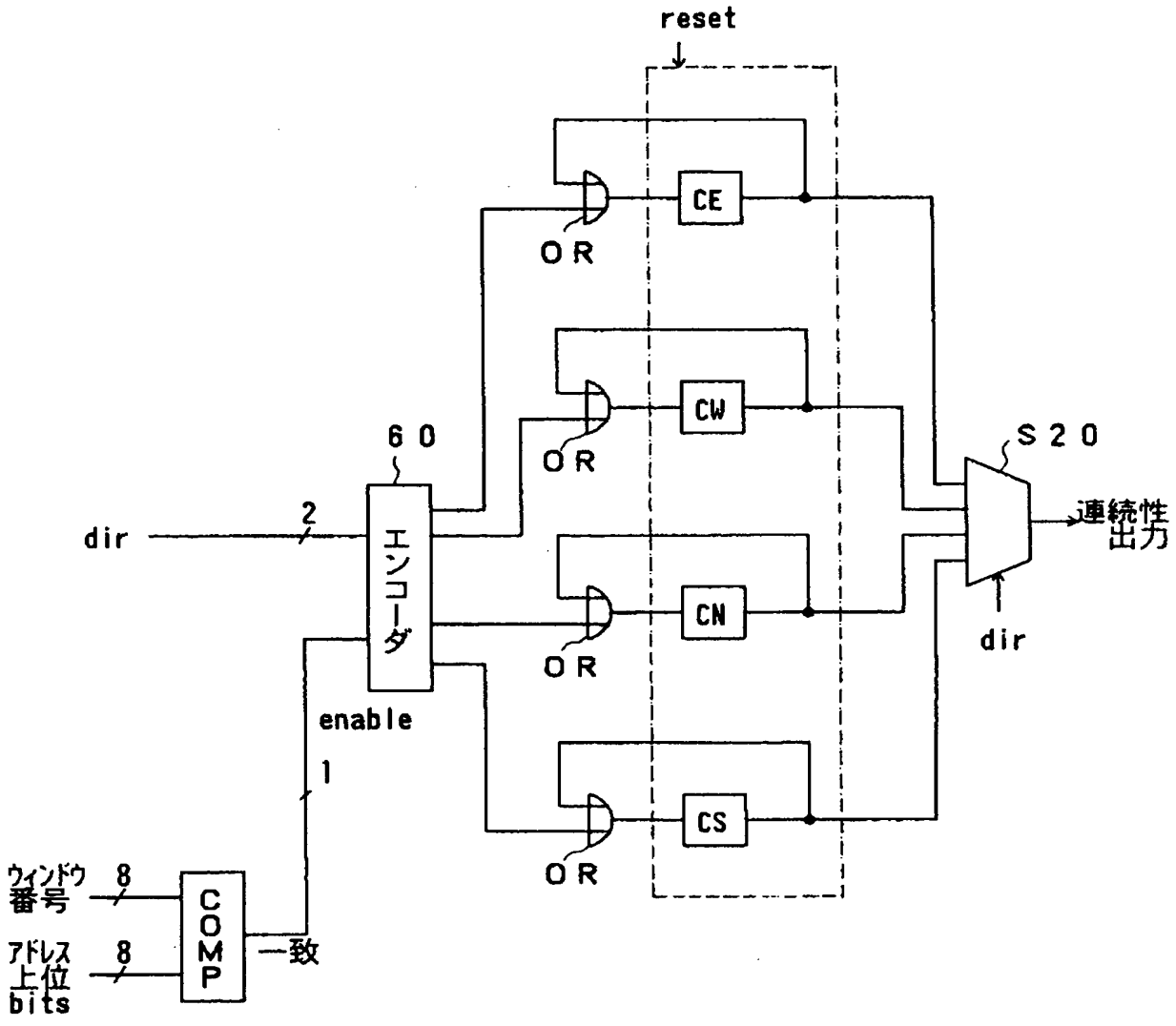
スケジューリング回路ブロック図

【第１０図】



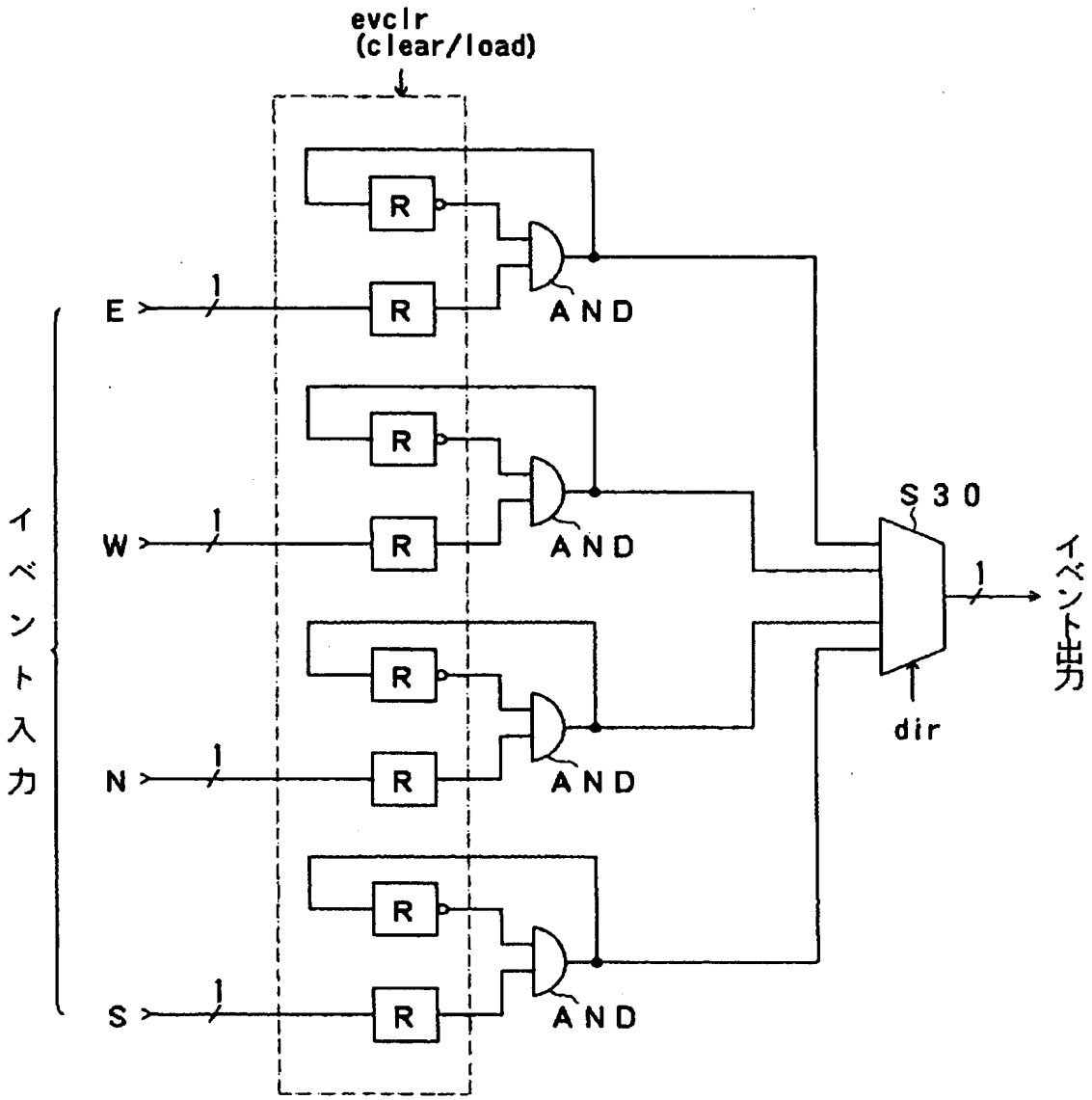
ウィンドウ番号入力部の例

【第１１図】



連　続　性　検　出　部　の　例

【第１２図】



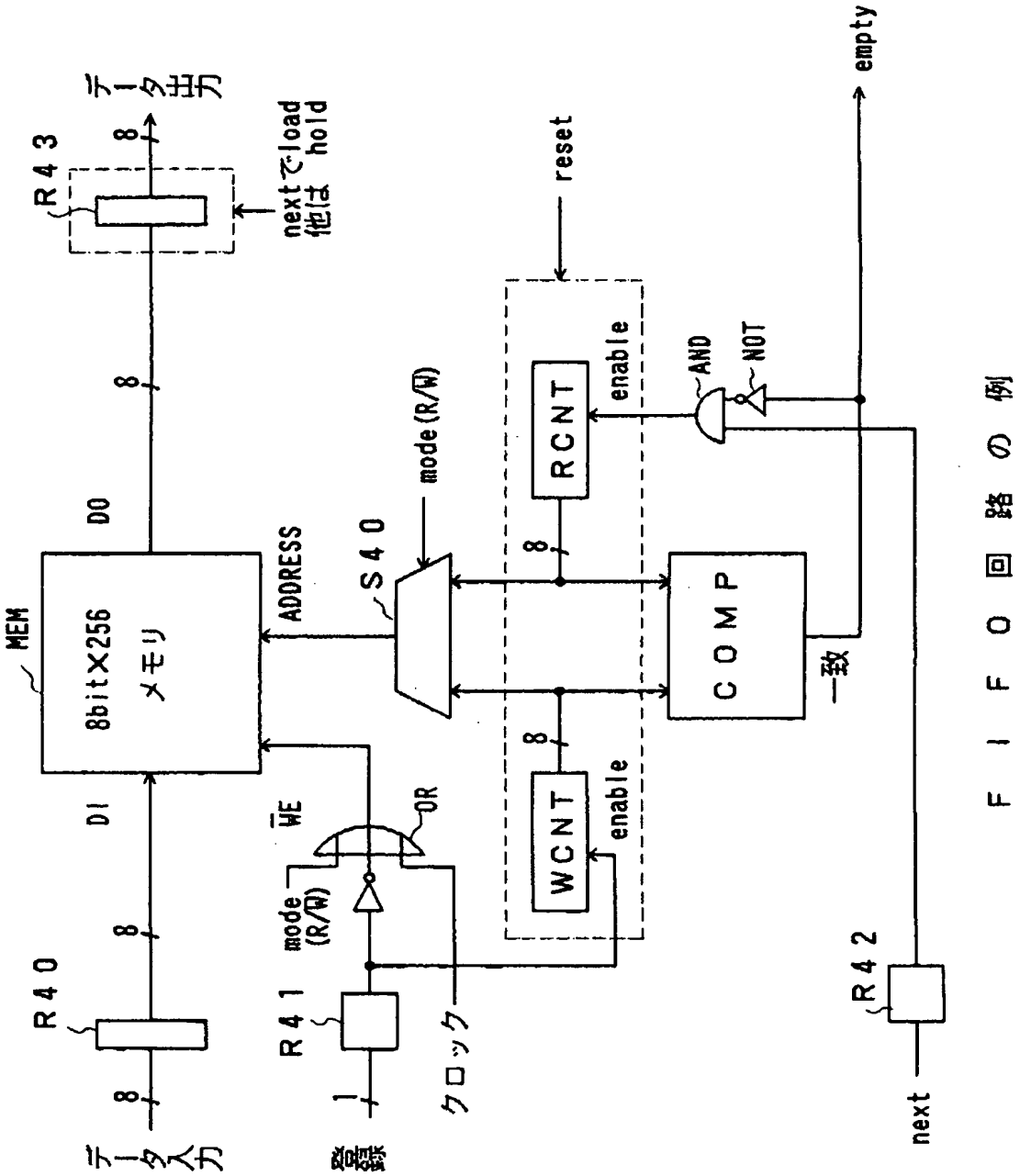
イベント入力部の例

【第１３図】

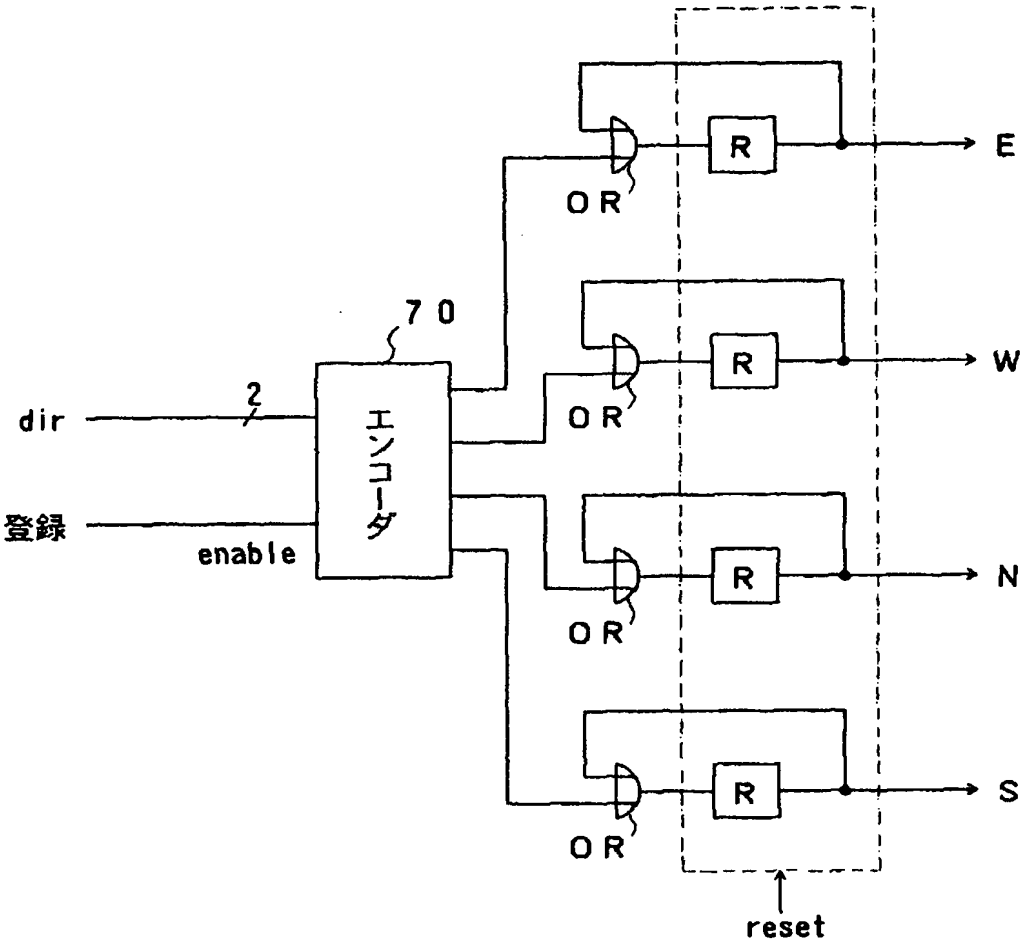
入力				出力	
登録テーブル出力	連続性出力	イベント出力	自己イベント	登録	アドレス保持
F	F	F	F	F	F
T	F	F	F	F	F
F	T	F	F	F	F
T	T	F	F		
F	F	T	F	T	F
T	F	T	F	F	F
F	T	T	F	F	T
T	T	T	F		
ー	ー	ー	T	F	T

イベント解釈回路論理説明図

【第１４図】

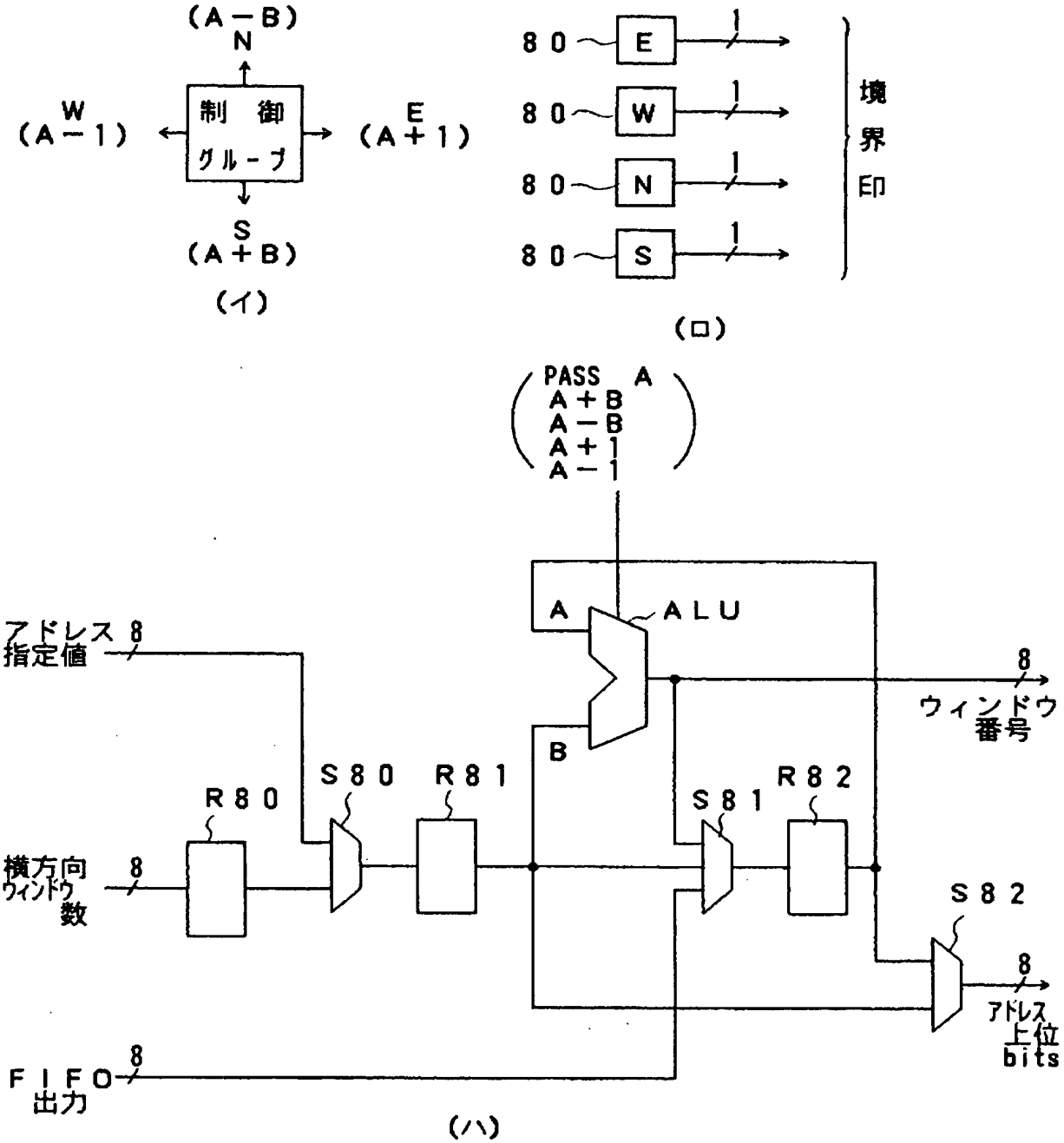

データ出力
8
R43
nextでload
他はhold
reset
empty
8
DO
mode(R/W)
RCNT
enable
AND
NOT
MEM
8bit×256
メモリ
ADDRESS
S40
8
COMP
一致
DI
WE
OR
8
WCNT
enable
mode
(R/W)
R40
R41
R42
8
1
データ入力
書込
クロック
next
FIFO回路の例

【第１５図】



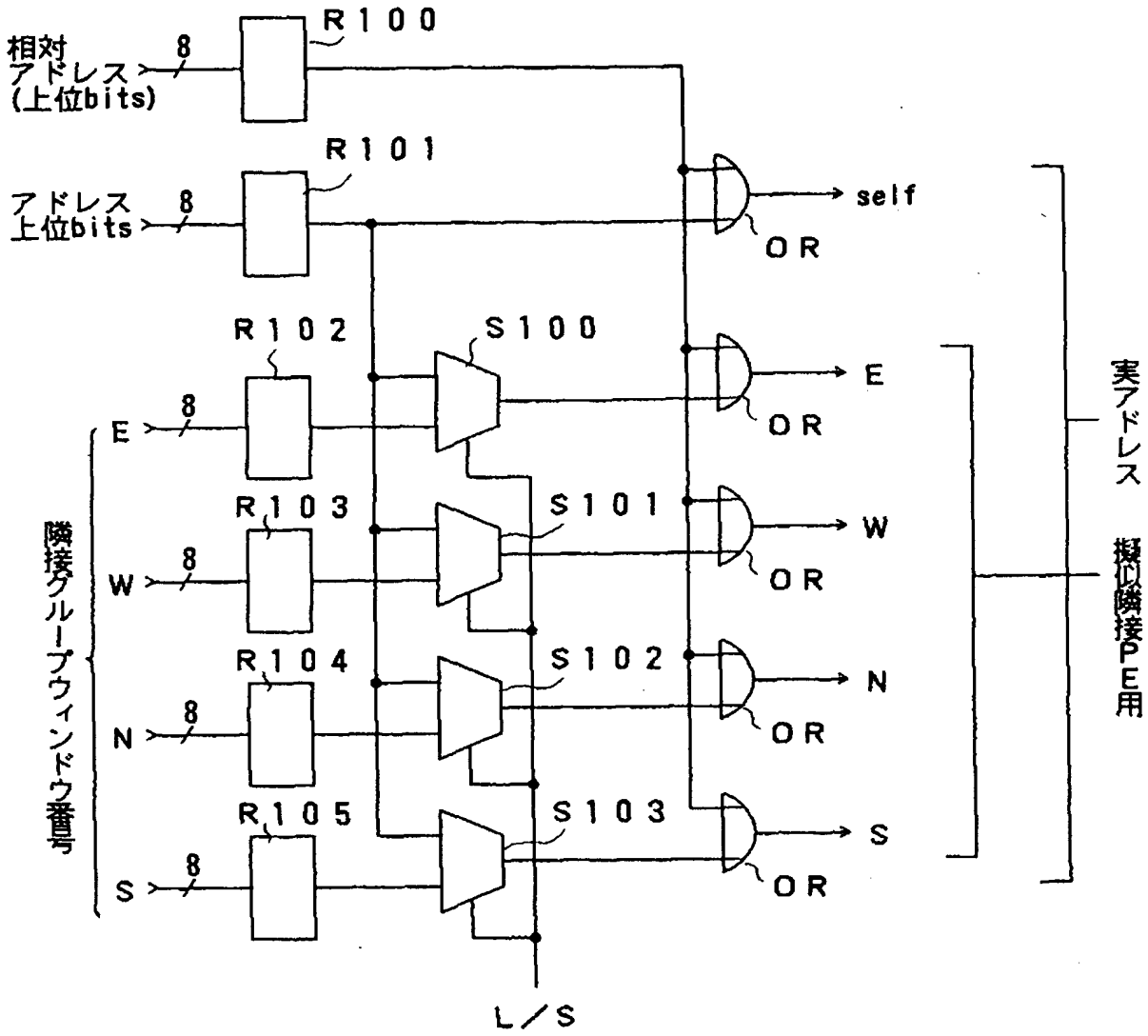
登　録　フ　ラ　グ　の　例

【第１６図】



アドレス計算回路説明図

【第１７図】



実アドレス生成回路の例

フロントページの続き

(72)発明者　澁谷　利行
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　三渡　秀樹
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内